Data Structure and Algorithm

Laboratory Activity No. 10

Intro to Graphs

|  |  |
| --- | --- |
| *Submitted by:* | *Instructor:* |
| Gabuyo, Ivan love D. | Engr. Maria Rizette H. Sayo |

OCT, 11, 2025

# Objectives

Introduction

A graph is a visual representation of a collection of things where some object pairs are linked together. Vertices are the points used to depict the interconnected items, while edges are the connections between them. In this course, we go into great detail on the many words and functions related to graphs.

An undirected graph, or simply a graph, is a set of points with lines connecting some of the points. The points are called nodes or vertices, and the lines are called edges.

A graph can be easily presented using the python dictionary data types. We represent the vertices as the keys of the dictionary and the connection between the vertices also called edges as the values in the dictionary.

A diagram of a triangle with green dots

AI-generated content may be incorrect.

Figure 1. Sample graph with vertices and edges

This laboratory activity aims to implement the principles and techniques in:

* To introduce the Non-linear data structure – Graphs
* To discuss the importance of Graphs in programming

# Methods

* 1. Discuss the following terms related to graphs:
     1. Undirected graph
     2. Directed graph
     3. Nodes
     4. Vertex
     5. Degree
     6. Indegree
     7. Outdegree
     8. Path
     9. Cycle
     10. Simple Cycle

# Results

 **Undirected Graph** – A graph where edges have no direction; the connection between nodes goes both ways.

 **Directed Graph** – A graph where each edge has a specific direction, often represented by arrows.

 **Nodes (Vertices)** – The points or objects that represent entities in the graph.

 **Vertex** – A single node in the graph.

 **Degree** – The number of edges connected to a vertex  
.

 **Indegree** – The number of edges coming into a vertex (in a directed graph).

 **Outdegree** – The number of edges going out of a vertex (in a directed graph).

 **Path** – A sequence of vertices connected by edges.

 **Cycle** – A path that starts and ends at the same vertex.

 **Simple Cycle** – A cycle that does not repeat vertices except for the start and end vertex.

Image

Figure 1 Screenshot of program

If an image is taken from another literature or intellectual property, please cite them accordingly in the caption. Always keep in mind the Honor Code [1] of our course to prevent failure due to academic dishonesty.

# Conclusion

Finally, this lab enhanced the understanding of graphs as non-linear data structure. The experiment successfully demonstrated how to represent the vertices and edges in python and how it's associated to derive logical relationships. Understanding how to manipulate graphs programmatically is an important skill for dealing with real world problems of networks, paths and connections. A simple text-generation activity like this also illustrated Python's power in manipulating data structures beyond the elementary linear ones such as stacks and queues.

**References**

Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2013). *Data structures and algorithms in Python.* Wiley.

Lafore, R. (2017). *Data structures and algorithms in Python.* Sams Publishing.

Python Software Foundation. (2024). *Python documentation: Data structures.* Retrieved from <https://docs.python.org/3/tutorial/datastructures.html>

Tanenbaum, A. S., & Bos, H. (2015). *Modern operating systems* (4th ed.). Pearson Education.